Rumor Detection on Social Media with Out-In-Degree Graph Convolutional Networks

Shihui Song*, Yafan Huang†, Hongwei Lu*
*Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering
†School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, China
{songsh, hyfshishen, luhw}@hust.edu.cn

***Abstract*— With the tremendous development in hardware computing and the widespread use of mobile terminal devices, there are increasingly more people who prefer to share their lives and opinions on social media. Though social media platforms allow everyone to express their opinions freely, they create convenience for rumor propagation in the meantime, which brings huge negative influence on the public and makes rumor detection extremely necessary. Currently, the most effective methods regard rumor propagation network as a graph and adopt graph convolutional networks (GCN) to detect rumor automatically. Such methods achieve promising performance in rumor detection, however, we argue that they have two critical defects: 1) they neglect the position contributions of rumor nodes in a graph, reducing the accuracy of rumor detection results; 2) they are inadequate in dealing with imbalanced data, which also indicates the inflexibility and the poor generalization ability of the model. To overcome these issues, we incorporate Katz centrality into spectral-domain graph convolution and propose a novel model named Out-In-Degree Graph Convolutional Networks (OID-GCN). Specifically, besides enhancing accuracy, Katz centrality can efficiently capture the position information of nodes, while the rest structure of OID-GCN shows a superb ability in dealing with imbalanced data. Comprehensive experimental results on two real-world datasets Twitter-15 and Twitter-16 demonstrate our OID-GCN outperforms existing methods.**

I. INTRODUCTION

As the internet applications develop continuously in recent years, the public's demands for social media have become more and more diverse, which indicates people nowadays prefer more interactive social media such as TikTok and Twitter rather than televisions and newspapers in the past. In such interactive social media, the interactions between a large number of users form a complex social network. This not only facilitates people's communication, but also provides convenience for creating and propagating rumors. Taking the research on COVID-19 tweets[1] from Northeastern University as an example, from January 2020 to September 2020, the number of COVID-19-related tweets was almost 30 million and the fake tweets rates was ranged from 6.96% to 61.00% on selected keywords. Such rumors not only caused mass panic among the people but also put huge pressure on public health organizations and other government departments. The World Health Organization (WHO) even established a website[2] to report misinformation online. Therefore, how to build a model to detect rumors quickly and accurately on social networks has attracted more and more research attentions.

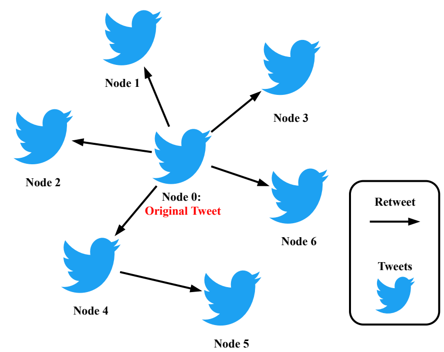


Fig. 1. A toy rumor propagation network to explain nodes' position contributions. Here *Node 0* is the original tweet (*i.e.* the rumor source node). As this node is retweeted by several other Nodes, it clearly holds a more important position. Also, *Node 0* has a greater contribution than other nodes (*e.g.* *Node 5*) to judging whether this rumor is true.

Traditional rumor detection models [1] are mainly based on statistic learning approaches with fine-grained feature engineering. However, we argue that they have a critical defect, that is, these models depend too much on manually-defined feature engineering and have difficulty in adapting to real-world social scenario. In order to reduce the manual preprocessing of datasets, methods based on deep neural networks [2]–[4], [8], [20] have widely adopted. Although these methods reach promising accuracy in dealing with embedded textual data without manually-defined features, they can not perform well in non-euclidean structure data such as rumor propagation network (*a.k.a.* graph).

In order to effectively represent rumor's propagation network and its textual information at the meantime, graph convolutional network (GCN) has been widely adopted recently. The core idea of GCN is to perform spectral-domain convolution operations on graph. More specifically, GCN aggregates features of target node's first-order neighbours to represent node information along with graph structure. Such method has a superb performance in embedding graphs, also, it makes significant progress in rumor detection task [5]–[7].

1. https://storybench.shinyapps.io/covid-tweets/

2. https://www.who.int/campaigns/connecting-the-world-to-combat-coronavirus/how-to-report-misinformation-online

Those GCN-based models do achieve promising results in several benchmark datasets, however, we argue that there exists two critical defects. (1) Existing GCN-based models **neglect the position contributions of nodes** in rumor propagation networks. As shown in Fig. 1, obviously, the original tweet by rumor source node (*i.e.* opinion leader) should be taken more into consideration than other ordinary retweets. Indiscriminately calculating these nodes will **reduce model's accuracy**. (2) In a rumor propagation network, there are both forwarding (*i.e.* retweet) and being forwarded relations, of which numbers vary greatly. Past GCN-based models are **inadequate in processing imbalanced data**. Therefore, their high accuracy was at the cost of **inflexibility**, making it hard to generalize to other scenarios.

In this paper, we mainly investigate two issues: (1) how to differentiate the position contributions of nodes in rumor propagation network, and (2) how to effectively process imbalanced data. To solve these, we propose an end-to-end rumor detection model named Out-In-Degree Graph Convolution Networks (**OID-GCN** for short). The contributions of this work are as threefold:

- We firstly adopt Katz centrality to differentiate the node position contributions in rumor propagation network to enhance the accuracy.
- We propose an end-to-end rumor detection model named OID-GCN for better processing imbalanced data.
- We conduct comprehensive experiments on two real-world datasets, Twitter-15 and Twitter-16, to demonstrate the accuracy and flexibility of our model.

II. RELATED WORK

In this section, we roughly classified existing rumor detection models to **non-graph methods** and **graph neural networks methods** and review some related works.

A. Non-Graph Methods

Most of previous non-graph methods can be divided to statistic learning methods and deep learning methods. In statistic learning methods [1], [9], [10], most of them focus on manually fine-grained feature engineering and incorporating external information to detect rumors. In order to reduce the cost of manual feature extraction, numerous deep learning models [2], [4], [11] were proposed to classify rumors through extracting features from original rumor information automatically. Ma et al [2] provided a Recurrent Neural Networks (RNN) based model for learning the hidden representations in time series. Recursive neural networks (RvNN) are also adopted [3] because of their strong ability in analyzing tree-structured data.

These models could capture features of content effectively, but ignore the total social network structure and the propagation structure, which plays an important role in the spread of rumors.

B. Graph Neural Networks Methods

Unlike deep learning models, Graph Neural Networks (GNN) (Battaglia et al [12]; Defferrard et al [13]; Hamilton et al [14]) could capture graph structural information better. After several fancy GNN-based models [16], [17] were firstly proposed, Bian et al [6] proposed a Bi-Directional Graph Convolutional Networks (Bi-GCN) to explore both top-down and bottom-up propagation information of rumors, while Wu et al [15] generated representation of nodes in a graph by a gated GNN-based algorithm and updated node information by exchanging its neighbours within limited time steps.

Though having taken graph structural features into account, we argue that existing GNN-based methods ignore the position contributions of nodes and can not efficiently process the imbalanced data.

III. PROPOSED MODEL

In this section, we give thorough descriptions of our proposed model OID-GCN, of which framework is shown as Fig. 2. To describe all the details clearly, we divide this section into 6 basic units. Firstly, we present notations used in this paper and formulate the rumor detection task. Then, we adopt Katz centrality to weigh the node features in an event. After that, we introduce how to define and construct out-in degree graphs and the network architecture of OID-GCN. Weighted concatenation and a fully-connected layer are also designed to obtain the predicted results. Finally, the optimization part can learn the parameters in OID-GCN.

A. Notations and Problem Formulation

Before elaborating our model, we firstly define the notations and formulate our task. Same as other rumor detection tasks, we let $E = \{e_1, e_2, ..., e_N\}$ denotes event set where e_i is the i-th event and N is the number of events. Every e_i here constructs a rumor propagation network, which is a directed graph and can be denoted as $e_i = \{r_1, n_2, n_3, ..., n_{N_i}\}$. Here r_1 is the rumor source node, $n_2, n_3, ..., n_{N_i}$ represent forwarded rumors (*i.e.* retweets) respectively, and N_i is the number of rumors in event e_i. Rumors are texts which are encoded by one-hot format. We regard each forwarded rumors as nodes in rumor propagation network, so that e_i's adjacency matrix can be represented as A_i, where its element $a^i_{mn} \in \{0, 1\}$ and $\forall m, n \in \{1, 2, ..., N_i\}$. In the meantime, we combine the embeddings of nodes $\{r_1, n_2, ..., n_{N_i}\}$ in rows into a node feature matrix X_i. Instead of previous bi-classification rumor detection task, we define our problem as a fine-grained four-classification task which is denoted as $\hat{y}_i \in \{NR, FR, TR, UR\}$. Specifically, $\hat{y}_i$ is the predicted results of e_i, NR, FR, TR, and UR represent **N**on-**R**umor, **F**alse **R**umor, **T**rue **R**umor and **U**nverified **R**umor respectively. In all, given an event e_i, our task can be formulated as the equation below:

$$\hat{y}_i = f_\Theta(e_i) \tag{1}$$

where $\hat{y}_i$ represents the predicted rumor class and f denotes the function of OID-GCN with its parameters Θ.

B. Katz Centrality Weighted

Katz centrality computes the relative influence of a node within a graph by measuring the number of first degree

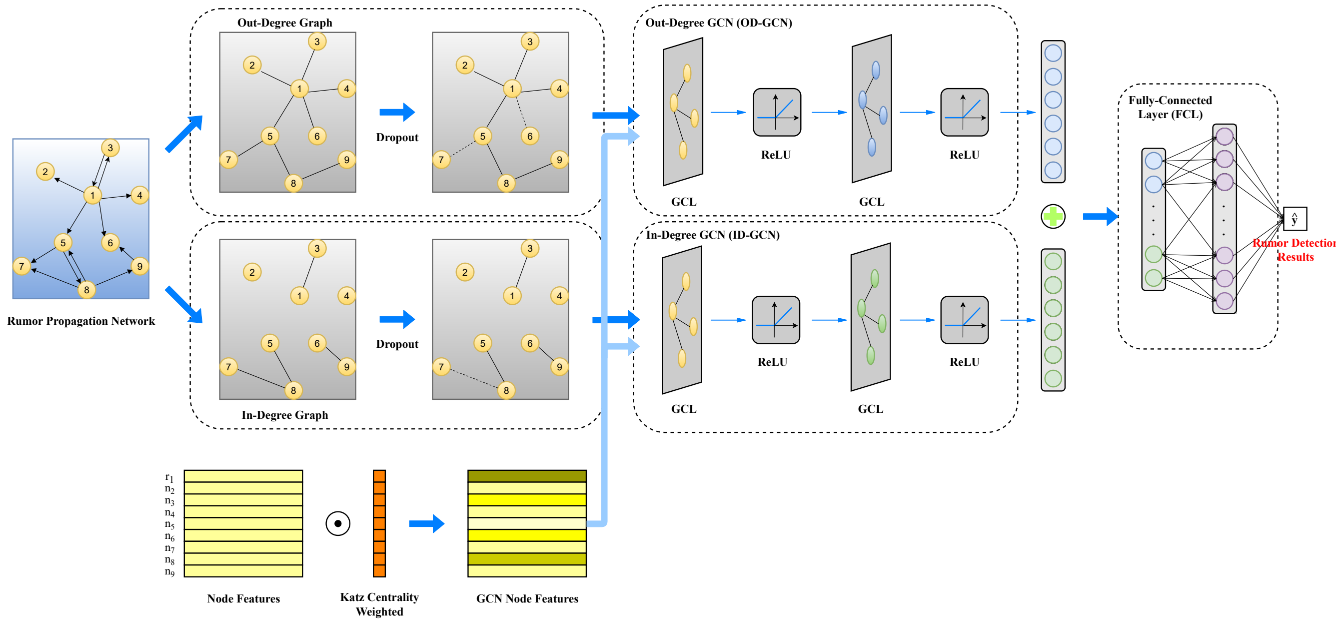


Fig. 2. The overall framework of the proposed model OID-GCN.

nodes. Compared with self-attention mechanisms computing the similarity weights of neighbour nodes [17], Katz centrality has a stronger ability to capture nodes' position information. Given an event e_i and its original adjacency matrix A_i, then mathematically:

$$\overrightarrow{C}_{\text{Katz}} = ((I - \alpha A_i^T)^{-1} - I)\overrightarrow{I} \tag{2}$$

where I is the identity matrix and $\overrightarrow{I}$ is a vector of size N_i. The value of attenuation factor α must be smaller than the reciprocal of absolute value of the largest eigenvalue of A_i. Numbers in $\overrightarrow{C}_{\text{Katz}} = \{kz_1, ..., kz_{N_i}\}$ denote the centrality of nodes in e_i. Thus the final Katz centrality weights can be calculated by a softmax function, as the equation below:

$$w_i = \text{softmax}(kz_i) = \frac{\exp(kz_i)}{\sum_{j=1}^{N_i} \exp(kz_j)} \tag{3}$$

The Katz weight set $\{w_1, w_2, ..., w_{N_i}\}$ will be then calculated with node feature matrix $X_i = \begin{pmatrix} r_1 & n_2 & \dots & n_{N_i} \end{pmatrix}$ as:

$$\tilde{X}_i = \begin{pmatrix} w_1 r_1 & w_2 n_2 & \dots & w_{N_i} n_{N_i} \end{pmatrix}^T \tag{4}$$

Note that operation $w_i n_i$ here is the scalar multiplication of vector, and $\tilde{X}_i$ will be the input feature matrix to the graph convolutional layer (GCL, *cf. Section* III-*D*).

C. The Construction of Out-In Degree Graphs

In rumor propagation network, the numbers of forwarding and being forwarded relations vary greatly, making data extremely imbalanced and hard to process. To resolve this and enhance flexibility of our proposed model, we divide the rumor propagation network (*i.e.* a graph) into an In-Degree Graph (OD-Graph) and an Out-Degree Graph (OD-Graph). Given the event e_i, it is easy to know its adjacency matrix $A_i \in \mathbb{R}^{N_i \times N_i}$ is asymmetric. Then the elements of OD-Graph A_i^O and ID-Graph A_i^I can be defined as following equation:

$$\begin{aligned} a_{mn}^{Oi} &= \begin{cases} a_{mn}^i & m \leqslant n \\ a_{nm}^i & m > n \end{cases} \\ a_{mn}^{Ii} &= \begin{cases} a_{mn}^i & m \geqslant n \\ a_{nm}^i & m < n \end{cases} \end{aligned} \tag{5}$$

Intuitively, OD-Graph is a symmetric matrix which copies upper triangular part of A_i and paste to its lower triangular part, while ID-Graph does the same thing in the opposite direction. In real-world scenarios, rumors are always propagated from a source node to other nodes, which indicates it has a tree structure. The earlier a node propagates the rumor, the smaller index it will have. Take OD-Graph A_i^O as example, it can be inferred that A_i^O is the adjacency matrix of a undirected graph holding the rumor propagation from old nodes to new nodes (*i.e.* forwarding relation). Thus, we can conclude that OD-Graph contains more information on forwarding relation and ID-Graph contains more information on being forwarded relation. Practically, we can set hyper-parameters on these two parts to control the OID-GCN's biases to different datasets. Such can significantly enhance the flexibility of our proposed model.

D. Out-In Degree Graph Convolutional Network

Given an event e_i, after defining its OD-Graph A_i^O, ID-Graph A_i^I, and its Katz centrality weighted feature matrix $\tilde{X}_i$, we then introduce the graph convolution structure of OID-GCN to better exploit the semantics from rumor propagation network. As shown in Fig. 2, the structures of OD-GCN and ID-GCN are similar. Both of them are composed of two layers of a graph convolution layer (GCL) and a rectified linear unit (ReLU) activation function. We also adopt Dropout in preprocessing A_i^O and A_i^I before performing the graph convolution operation. The edges in them are discarded at the ratio of p to overcome the overfitting issue. For simplicity, we still use symbols A_i^O and A_i^I in the later part of this section. In OD-GCN, the operation of a single GCL can be then formulated as:

$$\mathrm{GCL}_k(A_i^O, \tilde{X}_i) = \tilde{D}_i^{O-\frac{1}{2}} \tilde{A}_i^O \tilde{D}_i^{O-\frac{1}{2}} \tilde{X}_i W_k \tag{6}$$

where $\tilde{A}_i^O = A_i^O + I$ (*i.e.* adding self-loop), $\tilde{D}_i^O$ is the degree matrix of $\tilde{A}_i^O$, and W_k is the weight matrix to transform the dimension in k-th GCL. In all, the OD-GCN and ID-GCN are shown as the following equations:

$$\begin{aligned} S_i^O &= \mathrm{ReLU}(\mathrm{GCL}_2(A_i^O, \mathrm{ReLU}(\mathrm{GCL}_1(A_i^O, \tilde{X}_i))) \\ S_i^I &= \mathrm{ReLU}(\mathrm{GCL}_2(A_i^I, \mathrm{ReLU}(\mathrm{GCL}_1(A_i^I, \tilde{X}_i))) \end{aligned} \tag{7}$$

By such methods, we can obtain two semantic vectors S_i^O and S_i^I of event e_i which represent rumor forwarding information and being forwarded information separately.

E. Weighted Concatenation and Fully-Connected Layer

In order to combine two semantic vectors S_i^O and S_i^I flexibly to deal with biases on different datasets, we use a weighted concatenation operation to get an overall semantic vector:

$$S_i = \beta S_i^O \oplus (1-\beta) S_i^I \tag{8}$$

where $\oplus$ indicates concatenation operation and $0 \leqslant \beta \leqslant 1$ is a hyper-parameter on deciding how much we exploit information from forwarding relations. We will discuss the influence of β in details in *Section*-IV. A fully-connected layer (FCL) is then performed to transform S_i to a target $\mathbb{R}^{1\times 4}$ dimension for fined-grained classification, which is formulated as below:

$$H_i = W_b^T \mathrm{ReLU}(W_a^T S_i) \tag{9}$$

where W_a and W_b are two coefficient weight matrices of first layer and second layer of FCL respectively. The next step is to obtain a softmax function to convert $H_i \in \mathbb{R}^{1\times 4}$ to the final prediction (*i.e.* a probability distribution):

$$\hat{y}_i = \mathrm{softmax}(H_i) \tag{10}$$

where the details about softmax function can be referred to equation (3).

F. Optimization

In terms of optimization, our goal is to minimize the losses between predicted probability distribution $\hat{y}_i$ and the ground truth y_i. Since we regard rumor detection as a fine-grained four-classification task, negative log likelihood loss function can be used to learn the parameters:

$$\mathcal{L} = -\frac{1}{N} \sum_{e_i \in E} y_i \log(\hat{y}_i) \tag{11}$$

where E is the event set and N is the number of events in it, y_i and $\hat{y}_i$ represent the ground truth and predicted results of event e_i respectively. Note that we use stochastic gradient descent (SGD) for efficiently learning and L_2 regularization is omitted in equation (11) for simplicity.

IV. Experiments

In this section, we conduct various experiments to validate the effectiveness and efficiency of our proposed OID-GCN. Firstly, we give a thorough description on our selected datasets Twitter-15 and Twitter-16. Then the detaild experimental settings including baselines, evaluation metrics, and parameter settings are presented. Finally, we verify our model with several baselines from accuracy and flexibility perspectives.

A. Dataset Descriptions

Same as previous works [3], [4], [6], we select Twitter-15 and Twitter-16 as our datasets in this work as they contain both rumor textual information and well-structured rumor propagation networks (*i.e.* events). To our knowledge, they are by far the most valuable datasets for research on real-world rumor detection, of which statistics are shown in Table I. Each rumor propagation network contain hundreds of nodes, in which the first node represent the rumor source. And they are labeled in four categories: **T**rue **R**umors (TR), **F**alse **R**umors (FR), **U**nverified **R**umors (UR), and **N**on-**R**umors (NR).

TABLE I
STATISTICS OF TWITTER-15 AND TWITTER-16

Statistic	*Twitter-15*	*Twitter-16*
# of posts	331,612	204,820
# of Users	276,663	173,487
# of events	1,490	818
# of **T**rue **R**umors	374	205
# of **F**alse **R**umors	370	205
# of **U**nverified **R**umors	374	203
# of **N**on-**R**umors	372	205
Avg. time length / event	1,337 Hours	848 Hours
Avg. # of posts / event	223	251
Max. # of posts / event	1,768	2,765
Min. # of posts / event	55	81

B. Experimental Settings

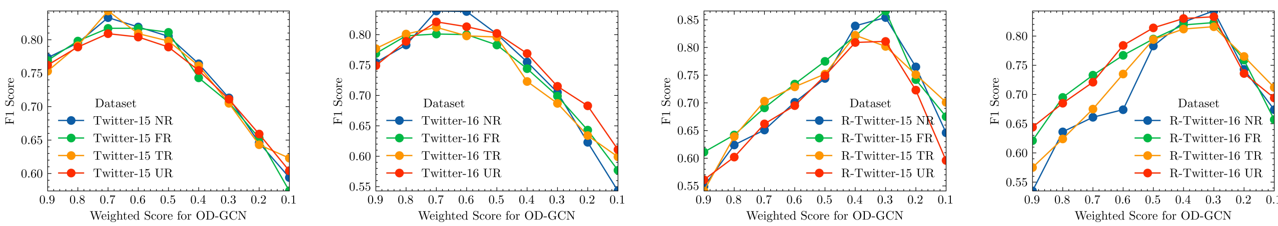
We compare our OID-GCN against the following baselines:

- **DTR** [10]: This is a decision-tree-based model to detect rumors via ranking the similar post cluster.
- **DTC** [1]: This key component of DTC is combining decision tree and manual feature engineering.
- **RFC** [18]: The is a random-forest-based classifier which is based on human-selected features from temporal, structural, and linguistic perspectives.
- **CNN-OM** [19]: This model is based on a two-channel CNN for sentence classification, which is a research field very close to rumor detection.
- **GRU-2 and LSTM-2** [2]: Two multi-layer rumor detection structures based on gated recurrent units (GRUs) and long-short term memory units (LSTMs) to capture high-level features between different time steps.
- **BU-RvNN and TD-RvNN** [3]: Two recursive neural networks (RvNN) based on bottom-up and top-down

TABLE II

AN OVERALL PERFORMANCE COMPARISON ON ACCURACY BETWEEN OUR OID-GCN AND SELECTED BASELINES. NR: NON-RUMOR, FR: FALSE RUMOR, TR: TRUE RUMOR, AND UR: UNVERIFIED RUMOR.

Dataset	*Twitter-15*					*Twitter-16*				
Method	Acc.	NR F1	FR F1	TR F1	UR F1	Acc.	NR F1	FR F1	TR F1	UR F1
DTR	0.409	0.501	0.311	0.364	0.473	0.414	0.394	0.273	0.630	0.344
DTC	0.454	0.733	0.355	0.317	0.415	0.465	0.643	0.393	0.419	0.403
RFC	0.565	0.810	0.422	0.401	0.543	0.585	0.752	0.415	0.547	0.563
CNN-OM	0.650	0.613	0.622	0.533	0.694	0.663	0.654	0.677	0.549	0.721
GRU-2	0.785	0.801	0.794	0.744	0.801	0.773	0.793	0.755	0.801	0.764
LSTM-2	0.733	0.711	0.745	0.624	0.761	0.752	0.749	0.766	0.733	0.747
BU-RvNN	0.708	0.695	0.728	0.759	0.653	0.718	0.723	0.712	0.779	0.659
TD-RvNN	0.723	0.682	0.758	0.821	0.654	0.737	0.662	0.743	0.835	0.708
TL-GCN	0.801	0.782	0.823	0.803	0.797	0.804	0.759	0.782	0.809	0.804
OID-GCN	**0.821**	**0.833**	**0.817**	**0.843**	**0.809**	**0.814**	**0.839**	**0.801**	**0.812**	**0.821**



(a) Twitter-15 (b) Twitter-16 (c) R-Twitter-15 (d) R-Twitter-16

Fig. 3. (a) and (b) analyze the influence of hyper-parameter β (*i.e.* how much we exploit information from OD-Graph) on model's accuracy, while (c) and (d) analyze the influence of hyper-parameter β on two new datasets R-Twitter-15 and R-Twitter-16.

traversal directions to exploit semantics from rumor propagation networks and textual information.

- **TL-GCN** [16]: This work innovatively proposes GCN and introduce it to node classification task. We here use a two-layer GCN (TL-GCN) as a baseline.
- **Bi-GCN** [6]: A state-of-the-art GCN-based rumor detection model constructed by rumor propagation and dispersion graphs. We compare OID-GCN with Bi-GCN to verify that our model can efficiently deal with data imbalance issues.

Among them, DTR, DTC, and RFC are statistic learning methods, CNN-OM, GRU-2, LSTM-2, and BU/TD-RvNN are deep learning methods, while Bi-GCN is a state-of-the-art graph-neural-network-based rumor detection model. We implement LSTM-2, TL-GCN and OID-GCN via pytorch. Accuracy (**Acc.**) and F1-score (**F1**) are adopted here as evaluation metrics for a fair comparison.

We firstly split the datasets to five parts randomly and leverage 5 cross-validation. In terms of parameter settings, we choose stochastic gradient descent (SGD) with a learning rate in range of {0.001, 0.01, 0.1, 0.3, 0.5}, and the coefficients of L_2 regularization are selected between $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. We embed all feature nodes to 64 dimension vector space. We also train the model within 200 epochs, which mostly achieves the best and stable results between 160 and 172 epochs. Noticed that any pre-trained parameters of our model are omitted for fairness.

C. Accuracy Comparison

Experimental results in this section are shown in Table II. Obviously, our proposed OID-GCN outperforms the other baselines in accuracy. Having considered all necessary conditions of rumor detection, including automatic detection, rumor propagation routines, and textual information, our OID-GCN efficiently boost overall Acc. and F1-score in every category. Note that here we use 70% information from OD-graph and the other 30% from ID-graph.

In order to further explore the influence of β (*cf. Section-III E*) on OID-GCN's results, we perform some supplementary experiments with β from 0.9 to 0.1. The results are shown in Fig. 3 (a) and (b). Almost every category achieves its highest accuracy with $\beta = 0.7$. In Twitter-15 and Twitter-16 datasets, the ratio between numbers of forwarding relations and being forwarded relations is more than 9, which corresponds to OD-Graph contains 8 times more semantics than ID-Graph. Under the condition that the Dropout ratios of OD-Graph and ID-Graph are 0.5 and 0.2, it is reasonable $\beta = 0.7$ has the best performance.

D. Flexibility Comparison

Due to the design of β, our model has a superb flexibility, which means it can generalize to different datasets. The results are shown in Table III and Fig. 3 (c) and (d).

TABLE III
RESULTS OF FLEXIBILITY COMPARISON.

Method	Acc.	NR F1	FR F1	TR F1	UR F1
R-Twitter-15					
TL-GCN	0.799	0.804	0.827	0.765	0.772
Bi-GCN	0.692	0.710	0.723	0.695	0.685
OID-GCN	**0.834**	**0.854**	**0.866**	**0.802**	**0.811**
R-Twitter-16					
TL-GCN	0.806	0.763	**0.824**	0.793	0.765
Bi-GCN	0.707	0.744	0.737	0.696	0.678
OID-GCN	**0.827**	**0.843**	0.823	**0.816**	**0.833**

We firstly reverse 50% of edges (*i.e.* forwarding relations) of original datasets and construct two new datasets R-Twitter-15 and R-Twitter-16. After such operation, the amount of information contained in OD-Graph and ID-Graph is basically balanced. Here all the other statistics remain the same. As shown in Table III above, since TL-GCN does not have any biases to graph structure and node features, the results of TL-GCN are very similar to before. Under the condition of $\beta = 0.3$, OID-GCN significantly outperforms the state-of-the-art Bi-GCN on every evaluation metrics.

As for the impact of β on this new scenario, we perform various additional experiments as illustrated in Fig. 3 (c) and (d). Almost every category achieves its highest performance with $\beta = 0.3$. Given that we reverse 50% of original edges' directions, the sparseness of newly constructed OD-Graph of ID-Graph is extremely close. It is convincing $\beta = 0.3$ achieves the best results as their Dropout ratios are the same as those in *Section*-IV C. Thus, we can conclude that β is efficient in adjusting the sparseness ratio between OD-Graph to ID-Graph. This also further proves our model's flexibility.

V. CONCLUSION AND FUTURE WORK

In this paper, we present an end-to-end model named OID-GCN for accurate rumor detection on Twitter. We firstly adopt Katz centrality to weight node feature matrix to better capture node position information. Also, we innovatively design an Out-In-Degree graph structure to flexibly represent rumor propagation process. Experimental results demonstrate that OID-GCN outperforms several state-of-the-art baselines in both accuracy and flexibility (*i.e.* generalization ability).

In the future, we aim to continue our research on rumor detection from following perspectives: though Twitter-15 and Twitter-16 are fine-grained datasets, the information in them is a bit out-of-date. COVID-19 and U.S. presidential election 2020 provide numerous topics and tweets, which can form extremely large and sparse social networks (graphs). It is very meaningful to clean up an up-to-date dataset and design more effective models for future research.

REFERENCES

[1] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in Proceedings of the 20th International Conference on World Wide Web, 2011, pp. 675–684.

[2] J. Ma, W. Gao, P. Mitra, S. Kwon, B. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 3818-3824.

[3] J. Ma, W. Gao, and K.-F. Wong, "Rumor Detection on Twitter with Tree-structured Recursive Neural Networks," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), 2018, pp. 1980-1989.

[4] Y. Chen, J. Sui, L. Hu, and W. Gong, "Attention-Residual Network with CNN for Rumor Detection," in Proceedings of the 28th International Conference on Information and Knowledge Management, 2019, pp. 1121-1130.

[5] M. Dong, B. Zheng, N. Q. V. Hung, H. Sun, and G. Li, "Multiple Rumor Source Detection with Graph Convolutional Networks," in Proceedings of the 28th International Conference on Information and Knowledge Management, 2019, pp. 569-578.

[6] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks," in Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 549-556.

[7] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang, "Rumor Detection on Social Media with Graph Structured Adversarial Learning," in Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020, pp. 1417-1423.

[8] F. Zhao, X. Gui, Y. Huang, H. Jin and L. T. Yang, "Dynamic Entity-based Named Entity Recognition Under Unconstrained Tagging Schemes," in IEEE Transactions on Big Data, 2021.

[9] K. Wu, S. Yang and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in Proceedings of the 31st International Conference on Data Engineering, 2015, pp. 651-662.

[10] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1395–1405.

[11] F. Yu, Q. Liu, S. Wu, L. Wang and T. Tan, "A convolutional approach for misinformation identification," in Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 3901–3907.

[12] P. Battaglia, R. Pascanu, M. Lai, D.J. Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in Proceedings of the 30th Annual Conference on Neural Information Processing Systems, 2016, pp. 3844–3852.

[13] M. Defferrard, X. Bresson, P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in Proceedings of the 30th Annual Conference on Neural Information Processing Systems, 2016, pp. 3844–3852.

[14] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Proceedings of the 31st Annual Conference on Neural Information Processing Systems, 2017, pp. 1024–1034.

[15] Z. Wu, D. Pi, J. Chen, M. Xie, and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," Expert Systems with Applications, 2020, Article 113595.

[16] T. N. Kipf, and M. Welling,"Semi-Supervised Classification with Graph Convolutional Networks," in Proceedings of the 5th International Conference on Learning Representations, 2017.

[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in Proceedings of the 6th International Conference on Learning Representations, 2018.

[18] S. Kwon, M. Cha, K. Jung, et al, "Prominent Features of Rumor Propagation in Online Social Media," in Proceedings of the 13th International Conference on Data Mining, 2013, pp. 1103-1108.

[19] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746-1751.

[20] Y. Huang, F. Zhao, X. Gui, and H. Jin, "Path-enhanced explainable recommendation with knowledge graphs" in World Wide Web, 2021.